

AlphaGo, developed by DeepMind, was the first AI agent to defeat a professional human player of go. Although IBM's Deep Blue was able to defeat Garry Kasparov, world champion of chess at the time, nearly 20 years earlier, AlphaGo is still an impressive accomplishment due to the relative complexity of go. Chess has an approximate average branching factor of 35 to go's 250, and similarly the approximate depths of chess and go are 80 and 150 respectively. This makes traditional search approaches like minimax ineffective in implementing go playing agents due to the vast size of the search space. Furthermore, the strategy of go is harder to express in straightforward terms, making it harder to incorporate distilled professional wisdom into algorithms to be utilized by computer agents.

In recent years, headway was made in go playing agents by using an alternative search strategy called Monte Carlo tree search (MCTS). MCTS works by applying Monte Carlo simulation to the tree search problem, simulating out potential outcomes of possible moves (i.e., child nodes of the current game state) by playing them out to completion with random legal moves. This establishes a decent probability of winning for the possible moves which are simulated, and is known to slowly converge upon the same decision tree as minimax even though it doesn't have a scoring function. However, in realistic play this would take too long to be feasible as the sample size would need to be quite large.

In order to guide the MCTS procedure, AlphaGo still needs a way to reduce the number of moves to consider so it can focus its efforts in deeply searching a few moves of high value. It utilizes two deep neural networks (DNNs), known as the policy network and the value network. The value network is used to estimate the value of a specific board state (essentially a scoring function, however one developed by training a deep neural network rather than using a well defined algorithm). The value network helps to focus the MCTS on moves that would likely be of high value. The other DNN, the policy network, is used to predict what moves were likely to be chosen. It was trained on a large corpus of professional play, and then through reinforcement learning against itself. The policy network helps focus the MCTS on paths that are of high likelihood of actually occurring. Additionally, AlphaGo is able to utilize parallel and distributed computing in order to perform deep searches under strict time limits per move.

By combining MCTS and its two DNNs, AlphaGo was able to achieve a very high level of play- outperforming all other known computer playing agents of go. Then, in a series of matches it was able to soundly defeat Fan Hui, a highly ranked and regarded professional player of go. Analysis of the way it plays has indicated that AlphaGo has developed unknown strategies that aren't currently part of the professional understanding of go.